MDC

MDC-ELLIPSOIDs

version 1.0

Minimum **D**istance **C**alculation between Ellipsoids described as implicit surfaces for rigid non-conformal contact detection.

Users and Developers Manual

1. How to install

- Download MDC-ELLIPSOIDs, unpack it to a preferred directory and run it in the MATLAB® environment by settling MATLAB®'s current folder to the MDC-ELLIPSOIDs root directory.

- The chosen directory is then added automatically to the MATLAB® path.

2. Directory content - MDC-ELLIPSOIDs/

The code package is composed by: 4 directories or modules, the main function, a demo file, a manual, and input data which can either be entered in a MATLAB® file or in an excel document.

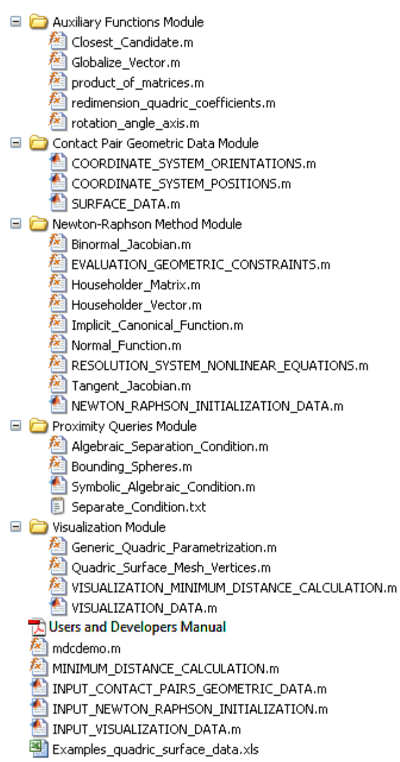

Figure 1 – MDC-ELLIPSOIDs folder content.

3. Running the demo file

- The demo file contains examples of the minimum distance calculation for 5 different contact pairs.

- Input data is available both in m-file (INPUT_CONTACT_PAIRS_GEOMETRIC_DATA.m, INPUT_NEWTON_RAPHSON_INITIALIZATION.m, INPUT_VISUALIZATION_DATA.m) and xls file formats (Examples_quadric_surface_data.xls). See next section for more on how to create input files.

- Type 'mdcdemo' on the command line window for a quick demonstration (e.g., >> mdcdemo).

- For each contact pair, the outputs are printed in the command window and the geometric data is rendered in a figure.

4. Inputs: m-file and excel file

There are three types of input data:

- **Contact pairs geometric data:** since the minimum distance calculation between two ellipsoids is a purely geometric problem, this type of input consist of affine transformation data (e.g., translation, orientation and dimension data) and the orientation of the normal vectors (inwards or outwards relatively to the surface). Identifiers for the contact pairs and surfaces are also required.

- **Newton-Raphson data:** the Newton-Raphson numerical method demands an initial approximation and a residual tolerance. To guarantee convergence, the initial approximation must be close to the solution. Note that the initial approximation does not have to, obligatorily, consist on points belonging to the surfaces. Such approximation can reside either inside or outside the surface.

- **Visualization data:** for visualization purposes, several may be the parameters that can be altered to better visualize the geometric data but here only the surface mesh resolution is of concern.

Table 1 lists all input data together with relevant information for each variable.

	Computational Variable	**Description**	**Dimension**	**Mathematical Variable**
Contact pairs geometric data	`cp_ID`	Contact pair ID.	1x1	-

	Computational Variable	Description	Dimension	Mathematical Variable
Contact pairs geometric data	surface_i_ID	Master surface ID.	1x1	-
	surface_j_ID	Slave surface ID.	1x1	-
	ntype_i	Normal vector orientation of master surface.	1x1	-
	ntype_j	Normal vector orientation of slave surface.	1x1	-
	r_Oalpha	Position of the reference system origin of rigid body α.	3x1	$\mathbf{r}_{\mathrm{O}\alpha}$
	r_Obeta	Position of the reference system origin of rigid body β.	3x1	$\mathbf{r}_{\mathrm{O}\beta}$
	r_alphai	Position of the reference system origin of surface i relatively to the reference system of rigid body α.	3x1	$\mathbf{r}_{\alpha i}$
	r_betaj	Position of the reference system origin of surface j relatively to the reference system of rigid body β.	3x1	$\mathbf{r}_{\beta j}$
	A_Oalpha	Orientation matrix of the rigid body α.	9x1	$\mathbf{A}_{\mathrm{O}\alpha}$

	Computational Variable	Description	Dimension	Mathematical Variable
Contact pairs geometric data	`A_Obeta`	Orientation matrix of the rigid body β.	9x1	$\mathbf{A}_{O\beta}$
	`A_alphai`	Orientation matrix of the surface i reference system relatively to rigid body α.	9x1	$\mathbf{A}_{\alpha i}$
	`A_betaj`	Orientation matrix of the surface j reference system relatively to rigid body β.	9x1	$\mathbf{A}_{\beta j}$
	`coeff_canon_i`	Semi-principal axes dimensions of surface i.	3x1	$\{a_i,b_i,c_i\}$
	`coeff_canon_j`	Semi-principal axes dimensions of surface j.	3x1	$\{a_j,b_j,c_j\}$
Newton-Raphson data	`qk`	Initial approximation (facultative).	6x1	$\mathbf{q}_{k=0}^{G}$
	`tolerance`	Residual tolerance.	1x1	-
Visualization data	`angle_resolution_i`	Equal angular discretization of the parametric expression of surface i.	1x1	-
	`angle_resolution_j`	Equal angular discretization of the parametric expression of surface j.	1x1	-

The input is organized as matrices in which each column corresponds to a contact pair:

- MATLAB® m-files
  - open INPUT_CONTACT_PAIRS_GEOMETRIC_DATA, enter data in the `CPSData` matrix;
  - open INPUT_NEWTON_RAPHSON_INITIALIZATION, enter data in the `NRMData` matrix;
  - open INPUT_VISUALIZATION_DATA, enter data in the `VISData` matrix.

- Excel file data sheets
  - access CONTACT_PAIRS_GEOMETRIC_DATA, enter data in the data sheet that will then be read as the `CPSData` matrix;
  - access NEWTON_RAPHSON_DATA, enter data in the data sheet that will then be read as the `NRMData` matrix;
  - access VISUALIZATION_DATA, enter data in the data sheet that will then be read as the `VISData` matrix.

Notes:

1. The orientation matrices are organized columnwise, i.e., columns are concatenated vertically, thus, the 9x1 dimension of the matrices;

2. In the excel file make sure to format the decimal numbers with a dot '.' or a comma ',' according to your excel program as MATLAB's `xlsread()` only reads one formatting type;

3. Choose by either entering numerical values for the initial approximation, `qk`, or by considering the estimate as the bisection between the surface centroids. In the latter case, insert '`NaN`' in each vector entry of `qk`.

5. Minimum distance calculation function

MDC-ELLIPSOIDs main function is named as MINIMUM_DISTANCE_CALCULATION and evaluates one contact pair at a time. Therefore, several contact pairs require several calls of the function. The inputs and outputs are the following:

Inputs

- `CPSData` – contact pair surface data;
- `NRMData` – Newton-Raphson method data;
- `VISData` – visualization data.

Outputs

- `min_d` – minimum distance between the ellipsoids of the contact pair;
- `qk` – minimum distance points in local coordinates;
- `q_G` – minimum distance points in global coordinates;
- `n_OP` – normal vector at point P;
- `n_OQ` – normal vector at point Q;
- `t_OQ` – tangent vector at point Q;
- `b_OQ` – binormal vector at point Q;
- `num_iter_NR` – number of iterations of the Newton-Raphson method;

- `t_NR` – elapsed time of the Newton-Raphson method;
- results of the proximity queries are printed at the command window;
- 3-D figure with the surfaces of the contact pair, the minimum distance points and the vectors that enter the system of non-linear equations.

The main function is composed by 5 parts. First, the input data is allocated to the corresponding variables. Secondly, proximity queries are evaluated but do not provided any geometric information regarding minimum distance. Thirdly, the system of non-linear equations is solved with the Newton-Raphson method where the Jacobian matrix was deduced as a analytical formula by vector calculus. Fourthly, the Euclidean distances of the four possible pairs of points that satisfy the common normal direction is verified to determine the unique solution. Finally, the geometric data is visualized in a 3-D figure.

For more information regarding any functions use the 'help' and 'doc' functions in the command window (e.g., >> `help mdcdemo`).

6. Expanding to other surfaces

To expand the code to other surfaces the following files have to be modified accordingly:

The equations are generic to any implicit surface that is C^2 continuous. Extending the methodology to other surfaces merely requires the deduction of the analytical expressions for $\mathbf{n}_{iP}$, $\mathbf{t}_{jQ}$, $\mathbf{b}_{jQ}$, $\left(\mathbf{n}_{iP}\right)_{\mathbf{q}^G}$, $\left(\mathbf{t}_{jQ}\right)_{\mathbf{q}^G}$, and $\left(\mathbf{b}_{jQ}\right)_{\mathbf{q}^G}$ which is usually a straightforward process, considering the analytical nature of the surface definition.

7. Jacobian matrix singularities

The Jacobian matrix becomes singular only when two or more rows or columns are linearly dependent, situation that includes rows or columns of zeros. A possible pitfall may occur when the following two conditions are, simultaneously, verified: $\mathbf{d}_{PQ} = \mathbf{n}_{OP}$ and $(\mathbf{d}_{PQ})_{\mathbf{q}} = (\mathbf{n}_{OP})_{\mathbf{q}}$.

8. License

MDC-ELLIPSOIDs is a collection of MATLAB functions to calculate the minimum distance between ellipsoidal surfaces.

Authors Acknowledgment
If you use MDC-ELLIPSOIDs in any program or publication, please acknowledge its authors by adding a reference to the Multibody Dynamics System paper:

D.S. Lopes, M.T. Silva, J.A. Ambrósio, and P. Flores, A mathematical framework for contact detection between quadric and superquadric surfaces, Multibody System Dynamics, 24(3): 255-280, 2010. DOI: 10.1007/s11044-010-9220-0

Commercial Use
The license above excludes every for-profit use of the code including, but non exclusively, development, testing, and certification of products or services. The Authors can release a license for commercial use, and the payment of a one-time fee. Please contact us for any information on this matter.

9. References

[1] D .S. Lopes, M.T. Silva, J.A. Ambrósio, and P. Flores, **A mathematical framework for contact detection between quadric and superquadric surfaces**, Multibody System Dynamics, **24**(3): 255- 280, 2010. DOI: 10.1007/s11044-010-9220-0

[2] A.S. Householder, **Unitary Triangularization of a Nonsymmetric Matrix**, Journal of the ACM, **4**(5):339–342, 1958.

[3] Y.-K. Choi, J.-W. Chang, W. Wang, M.-S. Kim, and G. Elber, **Continuous collision detection for ellipsoids**, IEEE Transactions on Visualization and Computer Graphics, **15**(2):311-325, 2009.

[4] A. Donev, S. Torquato, and F.H. Stillinger, **Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles II: applications to ellipses and ellipsoids**, Journal of Computational Physics, **202**:765-793, 2005.

[5] X. Lin and T.-T. Ng, **Contact detection algorithms for three-dimensional ellipsoids in discrete element modelling**, International Journal for Numerical and Analytical Methods in Geomechanics, **19**:653-659, 1995.

[6] S. D. Kwak, L. Blankevoort, and G. A. Ateshian, **A mathematical formulation for 3D quasi-static multibody models of diarthrodial joints**, Computer Methods in Biomechanics and Biomedical Engineering, **3**:41-64, 2000.

[7] S. Liu, C. C. L. Wang, K.-C. Hui, X. Jin, and H. Zhao, **Ellipsoid-tree construction for solid objects**, In: Proceedings of the 2007 ACM symposium on Solid and physical modeling, Beijing, China, 303-308, 2007.